HOUSE PRICE PREDICTION USING MACHINE LEARNING

INTRODUCTION

The real estate market plays a pivotal role in the global economy, with millions of transactions occurring annually. For both prospective buyers and sellers, as well as real estate investors, the ability to accurately predict house prices is of paramount importance. House price prediction can inform decisions related to buying, selling, or investing in real estate, and it can significantly impact the financial well-being of individuals and organizations. This introduces a compelling challenge in the field of data science and machine learning.

This research aims to explore the application of machine learning, data wrangling techniques, and neural networks to tackle the challenge of house price prediction. By leveraging these technologies, we seek to provide a reliable and automated solution for estimating house prices that takes into account a wide range of factors and provides predictions that are both accurate and adaptable to changing market conditions.

APPROACH

The real estate market is a complex and dynamic domain, making accurate house price prediction a challenging task. This study presents a comprehensive approach to address this challenge by leveraging machine learning, data wrangling techniques, and neural networks.

In this research, we will take a comprehensive approach to house price prediction, including data collection, data preprocessing and cleaning, feature engineering, model selection, neural network design, training, and evaluation. By the end of this study, we aim to offer insights into the most effective techniques for predicting house prices and their implications for the broader real estate industry. The results of this research may guide future developments in the real estate market and empower stakeholders with tools needed to make more informed decisions in this dynamic and valuable sector.

The key steps involved in our approach are as follows:

Data Collection: We gather a diverse dataset comprising various housing features such as square footage, number of bedrooms, bathrooms, location, and historical price trends. This data is essential for training and evaluating our models**.**

Data Cleaning and Preprocessing: Data wrangling techniques are applied to handle missing values, outliers, and categorical variables. This step ensures that our dataset is suitable for machine learning model training.

Feature Engineering: We engineer new features and perform dimensionality reduction to improve model performance and generalization. Techniques like one-hot encoding, feature scaling, and PCA are applied.

Model Selection: A variety of machine learning algorithms, including linear regression, decision trees, and gradient boosting, are evaluated for their ability to predict house prices. This step allows us to identify the most suitable model for our dataset.

Neural Network Architecture: We design a feed forward neural network that takes into account the engineered features as input. The network architecture includes multiple hidden layers with appropriate activation functions. We fine-tune hyper parameters to optimize the neural network's performance.

Training and Validation: The selected machine learning models and neural network are trained on a portion of the dataset and validated using a hold-out dataset or cross-validation techniques. Performance metrics such as mean squared error (MSE) and R-squared are used to assess model accuracy.

Prediction and Deployment: Once a model with satisfactory performance is identified, it is deployed to predict house prices for new and unseen data.

CODE ALGORITHM

Creating a complete house price prediction system using machine learning, data wrangling techniques, and a neural network involves several steps.

Here, I'll provide a high-level code algorithm outline to get started. We can implement this algorithm using a programming language such as Python with libraries like Scikit-Learn and TensorFlow/Keras for the neural network part.

By the algorithm given below we can adapt it to our specific dataset and requirements.

IMPORT NECESSARY LIBRARIES

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_squared_error

from sklearn.model_selection import GridSearchCV

from tensorflow import keras

from tensorflow.keras import layers
```

Step 1: Load the dataset

```
data = pd.read_csv("house_data.csv")  # Replace with the actual dataset file
```

Step 2: Data Wrangling and Preprocessing

Handle missing values

Encode categorical variables

Handle outliers

Feature engineering

Step 3: Split data into features (X) and target (y)

X = data.drop("house_price", axis=1)

y = data["house_price"]

Step 4: Split the dataset into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

Step 5: Feature Scaling

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)

Step 6: Model Selection (Random Forest Regressor)

rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

Step 7: Model Training

rf_model.fit(X_train, y_train)

Step 8: Model Evaluation

y_pred = rf_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)

print(f"Mean Squared Error: {mse}")

Step 9: Neural Network Architecture

model = keras.Sequential()

```
model.add(layers.Dense(128, activation="relu",
input_shape=(X_train.shape[1],)))

model.add(layers.Dense(64, activation="relu"))

model.add(layers.Dense(1)  # Regression task, so output is a single value
```

Step 10: Compile the Neural Network

```
model.compile(optimizer="adam", loss="mean_squared_error")
```

Step 11: Train the Neural Network

```
model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test,
y_test))
```

Step 12: Evaluate the Neural Network

```
y_pred_nn = model.predict(X_test)

mse_nn = mean_squared_error(y_test, y_pred_nn)

print(f"Neural Network Mean Squared Error: {mse_nn}")
```

Step 13:

Predict house prices using the trained models for new data

DATASETS

To build a house price prediction model, you'll need a dataset that contains historical data on houses, including features such as square footage, number of bedrooms, number of bathrooms, location, and, most importantly, the corresponding house prices. Here are a few popular datasets you can

use: Kaggle House Prices Dataset:

This dataset is available on kaggle and is a well-known choice for house price prediction projects. It contains a wide range of features and a large number of examples.

Link: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

California Housing Prices:

The California Housing Prices dataset is available in the scikit-learn library and is often used for educational purposes. It contains data related to housing prices in California.

You can access this dataset directly from scikit-learn using Python code.

python

```
from sklearn.datasets import fetch_california_housing

data = fetch_california_housing()
```

Zillow Prize Home Value Prediction:

Zillow, a real estate company, hosted a competition on kaggle with a large dataset for predicting home values. While the competition may have ended, you can still find the dataset on Kaggle.

Link: https://www.kaggle.com/c/zillow-prize-1

Boston Housing Dataset:

This is a classic dataset often used for regression tasks. It's available in the scikit-learn library and contains features related to housing in Boston, Massachusetts.

python

```
from sklearn.datasets import load_boston

data = load_boston()
```

Your Local Real Estate Databases:

Depending on your location, you might find real estate databases maintained by local authorities, real estate companies, or government agencies. These datasets can be valuable for predicting house prices in a specific area.

Websites and APIs:

Some real estate websites and APIs provide access to data on house listings, which you can use to create your dataset. Websites like Zillow, Redfin, or Realtor.com may have APIs or data scraping possibilities.

Before using any dataset, make sure you have the right to use it for your project, especially if you plan to share or publish your results. You should also clean and pre process the data to ensure it's suitable for machine learning.

Remember that the choice of dataset depends on your project's goals, the location you're interested in, and the specific features you want to include in your house price prediction model.

CONCLUSION

The study concludes by discussing the strengths and limitations of the proposed approach. It emphasizes the potential of neural networks improving house price prediction accuracy, especially when combined with effective data wrangling and feature engineering techniques.

In summary, this research demonstrates the effectiveness of combining machine learning, data wrangling, and neural networks to predict house prices accurately. The findings contribute to the advancement of real estate analytics and provide valuable insights for home buyers, sellers, and investors in the housing market.